The Existence of a Pseudo-triangulation in a given Geometric Graph

André Schulz*

Abstract

We show that the problem of deciding if a pseudo-triangulation is contained inside a geometric graph is NP-complete. For this we investigate the Triangulation Existence Problem, which is known to be NP-complete. We present a new proof for its NP-completeness and modify it in such a way that it can be applied for pseudo-triangulations.

1 Introduction

A *pseudo-triangle* is a polygon with exactly three convex corners. A planar partition of a point set into pseudo-triangles is called *pseudo-triangulation*. A pseudo-triangulation is called pointed, if all its vertices are incident to an angle greater than π. Many different applications for pseudo-triangulations are known.

The main focus of this paper lies on the problem, if there exists a pseudo-triangulation as a subset of a given geometric graph. The geometric graph does not have to be planar. We call this problem the Pseudo-Triangulation Existence Problem (PTRI).

The "triangulation version" of the PTRI is known as the Triangulation Existence Problem (TRI). Lloyd showed in 1977 [6] that the TRI is NP-hard by a reduction from CNF-SAT. Although the idea behind the construction is not difficult, it seems hard to modify the complex gadgets for new NP-completeness results.

In section 2 we will present a new proof for TRI. Instead of reducing from CNF-SAT we will use a reduction from Planar 3-SAT(which was not known in 1977). This allows us a simpler construction for the NP-hardness proof. We profit from the fact that Planar 3-SATis more structured than CNF-SAT. As a side effect it is now easier to modify the construction in such a way that we can apply it for PTRI.

Pseudo-triangulations do not always show the same behavior as triangulations (to name just one example, their flipping distance is smaller [1]). It is a natural question, in how far known results from triangulations can be generalized for pseudo-triangulations. This might help to understand the properties of pseudo-triangulations better.

*Institut für Informatik, Freie Universität Berlin, Germany, schulza@inf.fu-berlin.de

In [8] a NP-completeness result for triangulation (minimum vertex degree) was generalized for pseudo-triangulations with similar ideas used in this paper.

The problem of finding a pseudo-triangulation inside a triangulation was discussed in [7]. The complexity status of finding a pseudo-triangulation with minimal number of edges inside a triangulation was introduced as an open problem. Its status is still open.

2 A new proof for the NP-hardness of TRI

First of all we state the problem, for which we want to prove its NP-completeness

Triangulation Existence Problem (TRI)
Input: A geometric graph $G = (V, E)$.
Question: Is there a graph $G' = (V, E' \subset E)$ and G' is a triangulation ?

Theorem 1 *TRI is strongly NP-complete.*

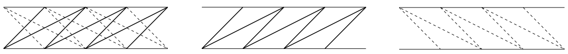
The proof of the theorem will be given by the following discussion. The Problem lies in NP, because we can verify in polynomial time if a guessed subgraph of G is a triangulation. To show NP-hardness, we reduce Planar 3-Sat to TRI. A formula ϕ is planar if it can be represented as a planar graph $G(\phi) = (V_\phi, E_\phi)$. The set V_ϕ is given by the variables and the clauses of ϕ. The pairs of all (negated) variables and their associated clauses define E_ϕ. The Problem if a planar formula in 3-CNF is satisfiable is known to be NP-complete [5].

The reduction from Planar 3-Sat is done by substituting edges and vertices of G_ϕ by more complex subgraphs (called *gadgets*). The resulting graph contains a triangulation, if and only if the formula is satisfiable,

We are using 4 different types of gadgets. The most essential gadget is the Wire gadget. It is responsible for carrying the value of a variable to the clauses and therefore it will be a replacement for the edges. The variables themselves are represented as a piece of the Wire gadget. To evaluate the clauses we introduce a Nand gadget and a Not gadget, which will be also used to negate variables. Finally we present a gadget which will split an edge, while maintaining the status of the wire for the outgoing parts. This gadget is called the Split gadget. Starting with the Wire gadget we will explain the gadget one by one.

Wire

The WIRE gadget represents the state of a literal which can be either true or false. Therefore it consist of a graph which contains exactly two triangulations (Figure 1). The two contained triangulations are called *black* (representing the value TRUE) and *dashed* (representing the value FALSE). We will also call their edges black and dashed. In the figures the dashed triangulation is drawn with dashed lines. The



(a) (b) (c)

Figure 1: (a) The WIRE (a); (b) & (c) its triangulations.

reader should check that it is not possible to switch inside the gadget from the dashed to the black triangulation and vice versa. This is due to the fact that there is no triangle which contains a dashed edge and a black edge. Therefore the choice of one edge determinates the status of all the other edges inside the gadget. It is possible to bend the gadget without destroying its structure. Figure 2 shows a 90 degree bend. A variable will be realized as a part of the WIRE

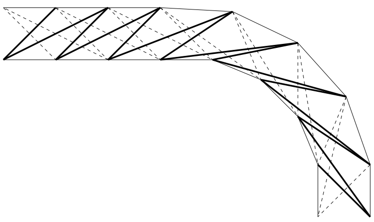


Figure 2: A WIRE with a 90 degree bend.

gadget (the part where we chose the fist diagonal).

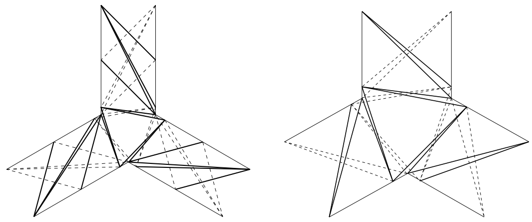
Split

The SPLIT gadget has three input parts (Figure 3). Like the WIRE it contains a dashed and a black triangulation. Since there is no triangle with a dashed and a black edge, it is not possible to switch between the black and dashed triangulation inside the gadget. For this reason the status of the wire is the same on the three output parts. The gadget will be used for producing multiple copies of a variable (or its negated version). Its open ends connect perfectly to the WIRE.

Not

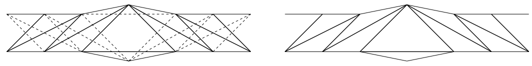
To realize a negation, we have to change the orientation of the diagonals inside the wire. Figure 4 shows how this can be done. Again we have two triangulations induced by the gadget and there exists no triangle with dashed and black edges. Hence, the orientation of the diagonals is switched by the gadget.

Nand



(a) (b)

Figure 3: (a) The SPLIT gadget; (b) a close up of the gadget.



(a) (b)

Figure 4: (a) The NOT gadget; (b) one of its triangulation.

The last gadget needed for the reduction is the NAND gadget. Its purpose is to evaluate the clauses (we can simulate an OR gate).

The NAND has three inputs and is slightly more complex than the other gadgets. It allows more than two triangulations. Therefore it contains edges which can not be handled as black or dashed (we call these edges gray). Lets assume that the dashed triangulation represents the value *true*. The gadget allows a triangulation for all possible input combinations, except when all are dashed triangulations. Lets have a closer look at the gadget shown in Figure 5. We

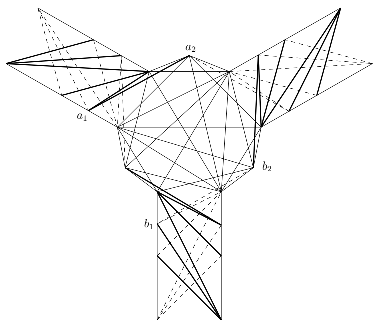


Figure 5: The NAND gadget.

see that three input wires meet in a 9-gon, which is filled with diagonals. The structure is symmetric under rotation by 120 degrees, but it is not symmetric by reflection. It can be observed that there are only three gray diagonals crossing the black edge a_1a_2 but there are 6 gray edges crossing the dashed diagonal b_1b_2. Hence having three dashed triangulated input wires makes it impossible to find a triangulation of the 9-gon. The removal of all gray diagonals in this setting leaves an empty hexagon (Figure 6.a), which can't be triangulated. On the other hand, all other combination of the input can be triangulated (as seen in Figure 6.b-d). All other combinations are symmetric versions of Figure 6. Therefore the functionality of a NAND gate is provided by the gadget and clauses can be evaluated in combination with the NOT gadget.

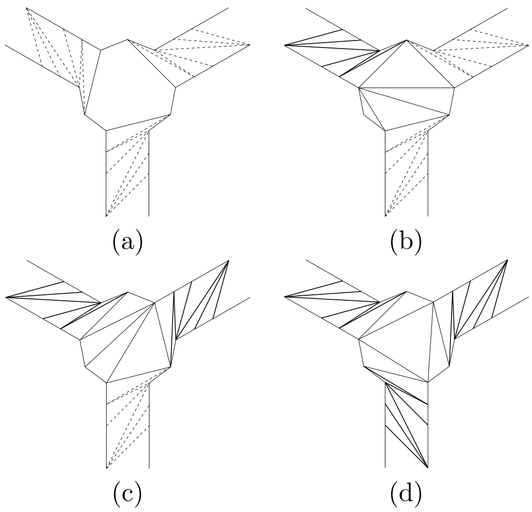


Figure 6: The NAND gadget with different input values.

After the substitution of the edges and vertices of G_ϕ by the gadgets, we might have pockets and holes inside the resulting graph. We will triangulate them arbitrarily. Any edge of the graph which is not crossed by any other edge has to be contained in the triangulation. Thus no boundary edge of a gadget can be deleted and the triangulation of the holes doesn't affect the functionality of the gadgets.

Clearly a formula ϕ is satisfiable, if and only if there exists a triangulation inside the constructed graph. To finishes the proof of Theorem 1 we observe that the reduction can be made in polynomial time.

3 Pseudo-triangulations inside a graph

The new proof of the NP-completeness of TRI allows us to attack similar problems with the same basic idea. One natural variation of TRI is the following:

Pseudo-Triangulation Existence Problem (PTRI)
Input: A geometric graph $G = (V, E)$.
Question: Is there a graph $G' = (V, E' \subset E)$ and G' is a pointed pseudo-triangulation ?

Theorem 2 *PTRI is strongly NP-complete.*

As done in the proof for Theorem 1 the proof will be given in the following discussion. Like in Section 2, we reducing again from PLANAR 3-SAT and we will introduce again the same set of gadgets (namely WIRE, SPLIT, NOT and NAND).

Wire

The WIRE gadget can be easily obtained from the one used for TRI. Now two pseudo-triangulations are part of the gadget. We call them again *dashed* and *black*. It is not possible to find a pseudo-triangle inside the gadget, which consists of a dashed and a black diagonal. Hence the choice of a diagonal determines the whole pseudo-triangulation. See Figure 7 for the corresponding pictures. It should be clear that bending

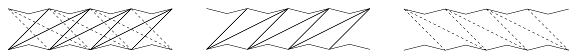


Figure 7: The WIRE and its pseudo-triangulations.

the gadget is no problem. We omit the picture for the 90 degree bend.

Not

The NOT gadget is basicly the same as the one for the triangulation case. We just ensure that every vertex contains an angle greater than π. Figure 8 shows the gadget.

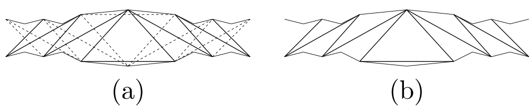


Figure 8: (a) The NOT gadget; (b) one of its triangulation.

Split

The SPLIT is shown in Figure 9.a. It has three incoming wire parts and a central area, which has to be covered by a pseudo-triangle. This is possible, if all

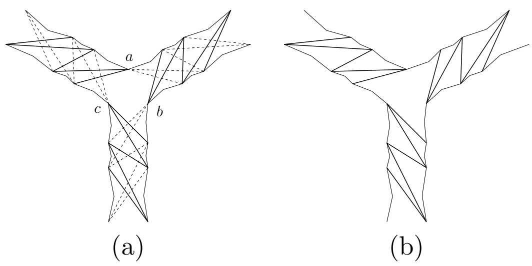


Figure 9: (a) The SPLIT gadget; (b) its black pseudo-triangulation.

three wires are dashed or black pseudo-triangulations (Figure 9.b shows the black pseudo-triangulation of the gadget). It is not possible to find a pseudo-triangulation for any other combination. In these case, we could not construct an angle greater π at at least one of the points a, b, c. Therefore the face covering the center of the gadget would be at least a pseudo-quadrilateral.

Nand

The NAND gadget (Figure 10) is based on the gadget used for TRI. Again we have a 9-gon and a number of gray diagonals. These have to be used to pseudo-triangulate the 9-gon. The pseudo-triangulation of

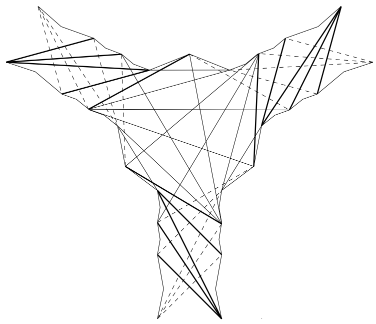


Figure 10: The NAND gadget.

the incoming wires forbids a certain set of diagonals, depending if the pseudo-triangulation is black or dashed. If all wires use dashed pseudo-triangulations it is not possible to pseudo-triangulate the 9-gon (Figure 11.a). All other cases (shown in Figure 11.b-d, up to symmetric equivalences) can produce valid pseudo-triangulations.

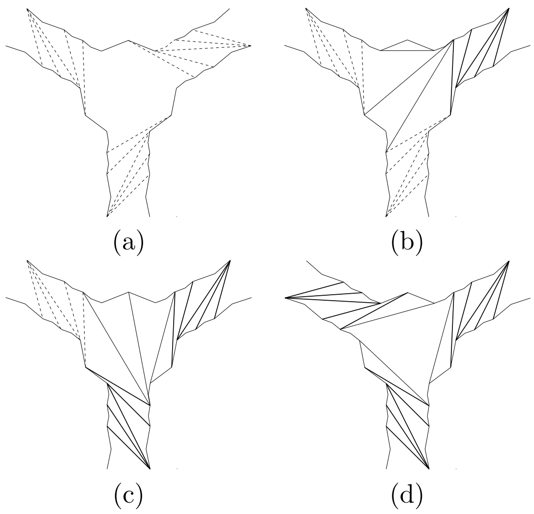


Figure 11: The NAND gadget with different input values.

The remaining holes of the constructed graph will be arbitrarily pseudo-triangulated (which is always possible). It remains to show that gadget boundary edges must be part of the pseudo-triangulation if one exists. If all vertices are pointed this follows from the fact that the removal of an edge, which is not crossed by any other edge, will construct a pseudo-quadrilateral. We leave the discussion for the non-pointed vertices of the gadgets to the full version of the paper.

It follows that this set of gadgets presents a valid reduction from PLANAR 3-SAT to PTRI.

4 Remarks and Open Problems

Since the gadgets for the reduction to TRI are small and easy to understand, they can be used to prove several similar NP-completeness results. One might think of the Problem if a quadrilateralization is contained inside given geometric graph.

Another interesting question related to PTRI is the following.

Planar Rigid Graph Containment (PRGC)
Input: Geometric graph $G = (V, E)$.
Question: Is there a planar (minimal) rigid graph $G' = (V, E' \subset E)$?

Although there are fast algorithms for testing planarity [4] and rigidity (e.g. [2]) it is not clear if we can find efficiently a rigid planar subset of a given graph. This problem is related to PTRI, since every pointed pseudo-triangulation forms a planar minimal rigid graph. Furthermore every planar minimal rigid graph can be embedded as a pointed pseudo-triangulation [3]. The gadgets we used will not help us, since a different embedding will destroy their functionality.

References

[1] S. Bereg. Transforming pseudo-triangulations. *Information Processing Letters*, 90(3):141–145, 2004.

[2] A. R. Berg and T. Jordán. Algorithms for graph rigidity and scene analysis. In *ESA*, volume 2832 of *Lecture Notes in Computer Science*, pages 78–89. Springer, 2003.

[3] R. Haas, G. Rote, D. Orden, F. Santos, B. Servatius, H. Servatius, D. Souvaine, I. Streinu, and W. Whiteley. Planar minimally rigid graphs and pseudo-triangulations. *Computational Geometry, Theory and Applications*, 2005.

[4] J. Hopcroft and R. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.

[5] D. Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*, 11(2):329–343, 1982.

[6] E. L. Lloyd. On triangulations of a set of points in the plane. In *Proc. 18th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 228–240, 1977.

[7] G. Rote, C. A. Wang, L. Wang, and Y.-F. Xu. On constrained minimum pseudotriangulations. In *COCOON*, volume 2697 of *Lecture Notes in Computer Science*, pages 445–454. Springer, 2003.

[8] A. Schulz. New results on pseudo-triangulations with low vertex degree. In *Proceedings of the 17th Canadian Conference on Computational Geometry (CCCG'05)*, pages 130–133, 2005.